Wildlife Recognition in Nature Documentaries with Weak Supervision from Subtitles and External Data

Aparna Nurani Venkitasubramanian[a,*], Tinne Tuytelaars[b], Marie-Francine Moens[a]

[a]*KU Leuven, Computer Science Department, Celestijnenlaan 200A, B-3001 Leuven, Belgium*
[b]*KU Leuven, ESAT-PSI, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium*

Abstract

We propose a weakly supervised framework for domain adaptation in a multi-modal context for multi-label classification. This framework is applied to annotate objects such as animals in a target video with subtitles, in the absence of visual demarcators. We start from classifiers trained on external data (the source, in our setting - ImageNet), and iteratively adapt them to the target dataset using textual cues from the subtitles. Experiments on a challenging dataset of wildlife documentaries validate the framework, with a final F_1 measure of approximately 70%, which significantly improves over the results of a state-of-the-art approach, that is, applying classifiers trained on ImageNet without adaptation. The methods proposed here take us a step closer to object recognition in the wild and automatic video indexing.

Keywords: Wildlife recognition, Cross-modal alignment, Domain adaptation, Multi-label classification, Incremental learning

1. Introduction

The dawn of the information age has seen tremendous growth in data especially in videos, making it increasingly challenging to facilitate quick and easy access to the relevant content. Currently, retrieval of 'relevant' videos is mostly based on user-tags. Not only are these tags often assigned in an ad-hoc manner, the process of acquiring them is also very cumbersome. Searching within the video to identify a particular segment in the video is even more difficult, since user tags are usually not available at such fine level of detail. So, one has to manually scan the video to find a certain interesting segment.

One possible solution is to automate the indexing process, by recognizing objects or actors shown in the video and then assigning labels. The subtitles or transcripts often present in a video provide cues to derive these labels [1, 2, 3]. However, solutions proposed in the literature use a visual demarcator such as a bounding box obtained from a face detector. Moving on to the problem of recognizing animals in wildlife documentaries [4], with the current state-of-the-art, it is not feasible to train a sufficiently accurate animal detector, since the variety within the bounding boxes is too large. Acquiring these bounding boxes by hand is tedious. Therefore, unlike [4], we are interested in a more realistic scenario where the bounding boxes are not available. In the absence of bounding boxes, the problem becomes much more challenging due to the following key issues - First, the presence of an animal is not known. Second, if the frame has animals, there could be multiple animals of possibly different species. Third, there are no ready examples that indicate with a reasonable confidence that a name-animal pair is linked. Fourth, isolating multiple animals cannot be easily done. Further, in this context, subtitles only provide weak cues, as they are not meant to describe the image content but rather give additional information to the viewer. This is

*Corresponding author: aparna.venkit@gmail.com

These are **south american sea lions** off the coast of patagonia.
They can't give birth while swimming, as **whales** and **dolphins** do, but have to come ashore.
And here, in dense groups, moving awkwardly between land and sea,
they're a great temptation to any hunter that can reach them

Figure 1: An example of a frame with the corresponding subtitle

in contrast to the body of work on using image captions or video descriptions [5, 6, 7], where the two modalities, namely vision and text, are much closer to each other.

In this article, we propose a weakly supervised approach to accurately associate animals in the video with their names in subtitles in order to assign tags or labels to video frames. We approach this as a multi-label classification problem using cross-modal data. We start from classifiers trained on external data (the source, in our setting is ImageNet [8]) and iteratively adapt them to the target dataset, using textual cues from the subtitles. In particular, we exploit the co-occurrence of animal mentions (and their co-referring expressions) in the subtitles with the animals (in their natural habitat) shown in the video to derive the correct labels. We experiment with a series of wildlife documentary videos with subtitles, from the British Broadcasting Corporation (BBC).

Figure 1 shows a video key frame together with the subtitle. Our approach of annotating the animals in this key frame is as follows: First, from the subtitle, we observe that the frame could contain a sea lion, or whale, or dolphin, or their combinations, or possibly no animal. We assume that if an animal is present in the video, it is also mentioned in the subtitle (or at least the subtitle contains a co-referent to it). We checked this assumption, and found out it was violated only in two key frames in our corpus. Therefore, in this example, we are interested in three binary classifiers (that indicate presence or absence) one for each possible animal - sea lion, whale and dolphin. Since we do not have reliable examples in our dataset that indicate a link between a name and an animal, we rely on an external dataset such as ImageNet to learn what these three animals look like. Then, we apply these classifiers to our data. However, as we see in Section 6, a direct application of the classifiers yields poor results, as the data distribution in the test (target) domain is very different from that of the training (source) domain [9]. Therefore, we propose to adapt the classifier learned on ImageNet to our dataset in an iterative manner. The basic idea of the adaptation is to exploit the co-occurrence of visually similar patterns (in the target dataset) with the names in the subtitles. To be able to count co-occurrence of the visually similar patterns with the text, we need a mechanism for grouping visual patterns. An obvious choice would be clustering, but clustering of these frames will be extremely noisy (as we show experimentally in Section 6). Therefore, we propose an alternative.

Li et al. [10] have shown that the Convolutional Neural Net (CNN) features (i.e., activations of a fully connected layer of a pretrained Convolutional Neural Network) used here have two interesting properties: 1) the features preserve their essence even after binarization and 2) they can be treated independently along the dimensions. We argue that these properties facilitate not only pattern mining of images as done in [10], but also allow *individual features (i.e, CNN activations) to be viewed as distinct elements depicting the existence (or non-existence) of some aspect of the image.* We can, therefore, represent an image with binarized CNN activations, and think of them as indicating the presence or absence of some aspect of the image. This is an intuitively appealing representation - using this representation, we can measure how the presence (or absence) of an animal label contributes to the presence (or absence) of a visual feature. This is measured by the probability of the feature given the animal name, initially using an external labeled dataset. Further, the independence property of the CNN features allows us to combine the probabilities of different features for the animal name in a Naive Bayes construction to obtain the likelihood of the name for the frame. In turn, the likelihoods of the names for the frame can be used to re-estimate the probabilities of different features for the animal name, effectively adapting to the target data. The process continues until convergence.

The rest of this paper is organized as follows: Section 2 discusses related work. Section 3 provides the background. Section 4 describes the general framework. Section 5 provides the implementation details. Section 6 discusses the experiments and Section 7 concludes the paper.

2. Related work

To the authors' knowledge, the problem of aligning animals from videos with their mentions in subtitles has not been studied apart from [4].

Animals are among the most difficult objects to recognize in images and videos, mainly due to their deformable bodies that often self occlude and the large variation they pose in appearance and depiction [11, 12]. Additionally, in the natural habitat, there are challenges due to camouflage and occlusion by flora. One of the earliest works on recognition of animals was that of Schmid [13], wherein models were constructed using Gabor-like filters and tested on different classes of animals with complex texture. Later, Ramanan et al. [14] proposed models to recognize animals using the shape and texture information in videos, built from a collection of segmented images. Berg and Forsyth [11] used textual and other cues such as color, texture and shape to generate visual exemplars of various classes of animals. Apart from these works that focus specifically on animals, there is a large literature on generic object detection. These methods are often evaluated on the Pascal VOC challenge dataset [15] which among its 20 classes also includes 6 classes of animals such as cats, dogs, cows and horses. There are also datasets that focus on animals such as Caltech UCSD Birds [16] and Stanford Dogs [17]. In this work, we propose a rather generic framework using the features of [18], which are activations of a convolutional neural network, as pioneered in [19].

Recently, there has also been some work on alignment across modalities for recognizing people [2, 3, 20]. These approaches rely on the use of a face-detector. While there are face detectors available with reasonable accuracy, there are no such detectors that allow localizing animals. In fact, not being able to localize the animals complicates the problem in multiple ways. Not only does background information and image clutter affect the visual descriptors when bounding boxes are unavailable, but also the many images that do not contain an animal at all can no longer be rejected upfront.

There has also been considerable interest in sentence/caption generation from images [5, 6, 7]. These approaches are not directly applicable to our setting: first, we have too few data to train similar models. Second, in our context, the subtitles and the visuals are not parallel, but complementary. For example, often a few animals are mentioned in the text, while the connected frame only shows one of them. The connection between the vision and the text is therefore much weaker.

This work draws on the principles of domain adaptation. Most works on domain adaptation are studied in the textual domain [9, 21, 22]. Lately, domain adaptation has also been gaining interest in computer vision [23, 24, 25]. Domain adaptation in an iterative context using a Naive Bayes classifier combined with an EM algorithm is also seen in [22]. In [22], text classification is performed in a multi-class setting. However, since documents and images can belong to multiple classes simultaneously, we address this problem from the perspective of multi-label classification.

The key contributions of this paper are as follows:

1. We propose an iterative framework for domain adaption in a multi-modal context for multi-label classification.
2. Exploiting two interesting properties of CNN features, namely 1) the features preserve their essence even after binarization and 2) they can be treated independently along the dimensions, we propose a feature transformation that allows us to split an image into components, and represent the image in terms of presence or absence of components. This transformation is beneficial since it allows association of the presence (or absence) of a component with the class labels, avoiding the need for the object detection step.

3. Background

We have a wildlife documentary with subtitles. On the visual side, we derive key frames $\mathbf{F} = \{f_1, f_2 \dots f_q\}$ from which we extract visual features with a suitable representation $\mathbf{A} = \{\mathbf{a_1}, \mathbf{a_2} \dots \mathbf{a_q}\}$. In general, these key frames may or may not contain animals. On the textual side, from

the subtitles, we extract the sentences. From these sentences, we identify the *unique* animal mentions or animal names $\mathbf{N} = \{n_1, n_2 \ldots n_p\}$.

Associated with every frame f_i, $1 \leq i \leq q$, we have a set $\mathbb{N}_i \subset \mathbf{N}$ of possible animal names derived from 5 subtitles to the left and right of the frame. The set $\mathbb{N}_i$ refers to the set of unique animal names derived from their mentions and coreferences in the subtitles. It is possible that the frame has some or all or none of the animals in $\mathbb{N}_i$. Corresponding to every name $n_l \in \mathbb{N}_i$, we have a binary label y_l indicating the presence or absence of n_l. Our objective is to find the most likely value of y_l corresponding to name n_l for every frame.

The problem of associating names to frames with manually annotated bounding boxes has been studied in [4]. As a baseline, we start with a straightforward extension of the same approach to entire frames. The basic idea is as follows - Group visual features representing frames across the whole video with a standard clustering approach such as k-means clustering. Start with an initial assumption that all the unique names are equally likely for every cluster. Iteratively refine using an EM algorithm, the likelihood of the names for the clusters, based on the co-occurrence frequency of the animal mentions with the elements of these clusters. Using the likelihoods, assign the best mapping between the animal names and the frames.

While good results were obtained with this approach when bounding boxes were available [4], applying it at the frame level is challenging due to the following key issues - first, clustering of the raw frames will be extremely noisy due to the parts of frame that do not contain animals. Note that a fuzzy-c-means clustering instead of the hard clustering will not suffice to overcome this problem. In fact, with a soft clustering, the noise from one cluster may get propagated to the other clusters. Second, it is not known if the frame contains any animal at all. Using the subtitle connected to the frame, one might conclude that the frame contains a certain animal, while the frame may in reality contain none. Under these circumstances, there are no good seed examples which indicate the possible visual representation of an animal. In the next section, we present a novel framework addressing these challenges.

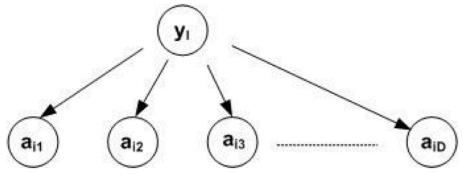

Figure 2: Generative model: the binary label y_l corresponding to name n_l generates the feature vector

4. General Framework

Our objective is to find the most likely value of y_l for every $n_l \in \mathbb{N}_i$ assigned to frame f_i, where $y_l = 0$ indicates the absence of the name n_l in that frame, while $y_l = 1$ indicates the presence of name n_l. Our approach is to train and iteratively adapt $|\mathbf{N}|$ classifiers, one for each name $n_l \in \mathbf{N}$. The rest of this section describes the procedure for each classifier.

4.1. Generative model

The probabilistic generative model for the data is shown in figure 2. We assume that every frame f_i is generated according to a probability distribution defined by a set of parameters θ_l, governing the label y_l. The likelihood of a frame is

$$p(f_i|\theta_l) = p(\mathbf{a_i}|\theta_l) = \sum_{y_l \in \{0,1\}} p(\mathbf{a_i}|y_l;\theta_l) * p(y_l) \qquad (1)$$

The above equation involves the term $p(\mathbf{a_i}|y_l;\theta_l)$ which denotes the probability of generating the frame given the label. In Sub-section 4.2, we describe how this term is computed, the parameter θ_l is defined in Sub-section 4.3.

The prior $p(y_l)$ allows to bring in other information, for example, dependencies based on picturedness [3] from text analysis, or background knowledge about the documentary (for example, likelihood of tigers might be low in a documentary about Africa). For simplicity, we use an uninformed prior. So, $p(y_l = 0) = p(y_l = 1)$. Eq. 1 then reduces to

$$p(f_i|\theta_l) = p(\mathbf{a_i}|\theta_l) \propto \sum_{y_l \in \{0,1\}} p(\mathbf{a_i}|y_l;\theta_l) \qquad (2)$$

Using Eq. 2, likelihood of all the data is

$$p(\mathbf{A}|\theta_l) \propto \prod_{f_i \in \mathbf{F}} \sum_{y_l \in \{0,1\}} p(\mathbf{a_i}|y_l;\theta_l) \qquad (3)$$

4.2. Naive Bayes model

The CNN features that we use here have properties that allow them to be treated along the dimensions independently [10]. This allows us to make the standard Naive Bayes assumption. The key idea here is that rather than viewing the frame in its entirety, the frame can be viewed as a collection of D features. Then, the term $p(\mathbf{a_i}|y_l;\theta)$, of Eq. 1 can be estimated as follows:

$$p(\mathbf{a_i}|y_l;\theta_l) = p(< a_{i1}, a_{i2}, \ldots a_{iD} > |y_l;\theta_l) = \prod_{v=1}^{D} p(a_{iv}|y_l;\theta_l) \tag{4}$$

Next, we describe how the probabilities $p(a_{iv}|y_l;\theta_l)$ of individual features for the label are computed.

4.3. Binarization

Yet another interesting property of the CNN features is that they can be binarized without losing the essence [10]. This property can be exploited to compute the probabilities of the features $p(a_{iv}|y_l;\theta_l)$. We make use of the fact that visually similar patterns co-occur with the same name. Instead of clustering, as done in [4] for bounding boxes and in our baseline for frames, we simply binarize the CNN featuress along each dimension, by splitting mid-way[1] between the minimum and maximum values of each dimension, over the entire data. The intuition behind the binning is as follows: we can represent an image with binarized CNN activations[2], and think of them as indicating the presence or absence of some aspect of the image. The binarization is intuitively appealing because with this transformation, it is easy to infer the association between the presence (or absence) of a feature and the presence (or absence) of a name.

$$p(a_{iv}|y_l;\theta_l) = p(\beta_v|y_l) \tag{5}$$

where $\beta_v \in \{0, 1\}$ is the bin to which a_{iv} belongs.

[1] We experimented with two alternatives to this equal width approach: 1) An equal frequency approach with a correction to ensure that if more than 50% of the values along a dimension are 0 (since we are dealing with sparse matrices), they should all belong to the same bin and 2) A rank-based approach where we set the r highest values along each dimension to 1 and the rest to 0. We experimented with different values of r and found that the equal width approach performed better than the equal frequency and rank-based approaches.

[2] We show in Section 6, that the binarization of features does not have a significant impact on the classification accuracy.

The parameter θ_l is a collection of bin probabilities $p(\beta_v|y_l)$ for name n_l along each dimension, where v indicates the dimension, β_v is the bin along dimension v.

4.4. Expectation-Maximization

For every bin β_v along dimension v and label y_l for name n_l, the parameters are initially estimated from an external reference dataset with labeled images

$$p(\beta_v|y_l) = \frac{freq(\beta_v, y_l)}{freq(\beta_v, y_l) + freq(\overline{\beta_v}, y_l)} \tag{6}$$

where $\overline{\beta_v}$ is the one's complement of β_v.

In the E-step, we estimate the posterior probability of the class labels, $p(y_l|\mathbf{a_i};\theta)$ by using Bayes' rule and applying a normalization.

$$p(y_l|\mathbf{a_i};\theta_l) = \frac{p(y_l)\prod_{v=1}^{D} p(a_{iv}|y_l;\theta_l)}{p(y_l)\prod_{v=1}^{D} p(a_{iv}|y_l;\theta_l) + p(\overline{y_l})\prod_{v=1}^{D} p(a_{iv}|\overline{y_l};\theta_l)} \tag{7}$$

Where $\overline{y_l} = 0$ if $y_l = 1$ and vice versa. Using Eq. 5, Eq. 7 can be written as follows:

$$p(y_l|\mathbf{a_i};\theta_l) = \frac{p(y_l)\prod_{v=1}^{D} p(\beta_v|y_l)}{p(y_l)\prod_{v=1}^{D} p(\beta_v|y_l) + p(\overline{y_l})\prod_{v=1}^{D} p(\beta_v|\overline{y_l})} \tag{8}$$

where β_v is the bin to which a_{iv} belongs.

In the M-step, new classifier parameters, θ_l, are re-estimated based on the current values of $P(y_l|\mathbf{a_i};\theta_l)$ as follows.

$$p(\beta_v|y_l) = \frac{\sum_i p(y_l|\mathbf{a_i};\theta_l) * m(\beta_v, \mathbf{a_i})}{Z} \tag{9}$$

where $m(\beta_v, \mathbf{a_i})$ is 1 if a_{iv} belongs to bin β_v and 0 otherwise. Z is a normalization constant to ensure $p(\beta_v|y_l) + p(\overline{\beta_v}|y_l) = 1$. These last two steps are iterated until convergence. Upon convergence, for every name n_l, the most likely label $y_l = 0$ or 1 is assigned to every frame. With this framework, it is possible that $y_l = 0;\ \forall n_l \in \mathbb{N}_i$ for a certain frame f_i. In that case, it will be predicted that the frame has no animal. This is interesting because there will be several key frames that do not contain any animal. The steps above are summarized in Algorithm 1.

5. Implementation details

This section describes the pre-processing of the textual and visual data, and the learning of animal classifiers from an external dataset.

Algorithm 1: The iterative framework to identify the animals in a frame

Input : Labeled set of images from ImageNet
Frames of the target dataset **F**
Possible names $\mathbb{N}_i$ for each frame f_i

$\mathbf{N} = \cup_i \mathbb{N}_i$
for every name $n_l \in \mathbf{N}$ **do**
  Initialize $p(\beta_v|y_l)$ from ImageNet, using Eq. 6
  while likelihood measured by Eq. 3 increases **do**
    /* E-Step: */
    for every frame $f_i \in \mathbf{F}$ **do**
      Estimate $p(y_l|f_i;\theta_l)$, using Eq. 8
    /* M-step: */
    for every bin β_v along dimension v **do**
      Re-estimate $p(\beta_v|y_l)$ using Eq. 9
for every frame f_i **do**
  for every name $n_l \in \mathbb{N}_i$ **do**
    Choose the label $y_l = \text{argmax}_{y_l} P(y_l|f_i)$

Output: Most likely values of y_l for every frame f_i

5.1. Pre-processing of the Text and Visual data

On the text, named-entity recognition and coreference resolution are performed as described in [4]. To analyze the video, shot cut detection and keyframe extraction are done using [26]. Subsequently, visual features are extracted using the powerful Convolutional Neural Networks (CNN) [19], which are deep structures comprising several layers of feature extractors. In particular, we use the CNN-M-128 architecture of [18], which is trained on 1,000 object categories from ImageNet [8] with roughly 1.2M training images. With this representation, the activities of the penultimate layer (7th fully connected layer) are used as features. This model yielded 128 features.

5.2. Learning from ImageNet

We use ImageNet to learn probabilities of the binarized features given the name. The process is as follows: For every unique animal name n_l, we collect a set $\mathbf{I}_{\mathbf{n_l}}$ of images from ImageNet. The set $\mathbf{I} = \cup_{n_l \in \mathbf{N}} \mathbf{I}_{\mathbf{n_l}}$ constitutes a dataset labeled with animals that we use for training animals classifiers.

We then train binary classifiers for each of these entities on the collection of relevant images from ImageNet, using a one-versus-rest scheme. For each unique animal mention n_l, the positive class comprises the images $\mathbf{I}_{\mathbf{n_l}}$ containing that animal, while the negative class includes all the other images, $\mathbf{I} - \mathbf{I}_{\mathbf{n_l}}$. On inspecting the data from ImageNet, it was found that there were very few examples with multiple species in the same image, so it is reasonable to assume that the negative class for an animal does not include that animal.

For all the images in **I**, we extract the CNN visual features trained on Imagenet [19] as before and binarize them. Once the bins have been computed, the probability of a bin β_v along the dimension v for a label y_l is estimated by counting the co-occurrence of the name with the bin, using Eq. 7.

6. Experiments and Results

The data used in our experiments is the DVD Great Wildlife Moments[3] from the BBC. This is an interlaced video with a duration of 108 minutes at a frame rate of 25 frames per second, and the frame resolution is 720x576 pixels. The video consists of 28 chapters and all the chapters except the ones containing just one animal are evaluated. This leaves us with chapters 14 to 28. Applying shot cut detection [26] on these chapters, we obtained 602 key frames. Of these, 302 frames had no animal. The remaining 300 contained 365 animals in total. We run our algorithm on all the 602 frames.

The subtitles are distributed throughout the video and contain a total of 7,304 words in 545 sentences. 186 animal names are mentioned in these subtitles. The distribution of animal species over the key frames is shown in figure 3. The number of animal mentions associated with each frame over the entire dataset is also shown in Figure 3. Note that based on the subtitles, there are several frames that have at least 2 names associated. On the visual side, however, there are several frames that do not contain any animals. This shows the ambiguity in text and vision.

The evaluation of the text pre-processing is as in [4]. In order to evaluate our algorithm, we first consider a set of approaches purely based on vision, using an external dataset (Section 6.1). Second, we consider a baseline entirely based on text (Section 6.2). Third, we report the

[3] http://www.bbcshop.com/science+nature/great-wildlife-moments-dvd/invt/bbcdvd1131/

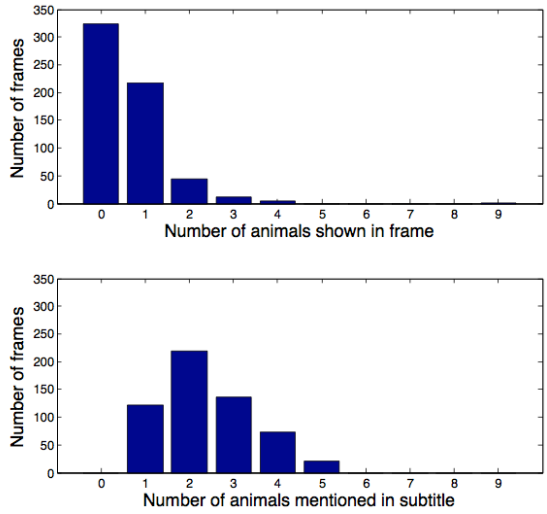

Figure 3: Distribution of animals over the key frames

results of state-of-the-art approaches based on clustering and understand their shortcomings in our scenario where bounding boxes are absent (Section 6.3). Next, we study the impact of binarization from two perspectives- a) as an alternative feature representation and b) as a means of grouping (Section 6.4). Finally, we evaluate our pipeline and show the value of the iterative learning (Section 6.5). Table 1 shows the results of our approach compared to various other approaches.

Precision and recall are computed over the entire dataset as follows:

$$precision = \frac{number\ of\ correct\ guesses}{total\ number\ of\ guesses} \quad (10)$$

$$recall = \frac{number\ of\ correct\ guesses}{actual\ number\ of\ animals\ present} \quad (11)$$

Figure 4 shows an example of our approach in the realistic scenario without bounding boxes.

6.1. How good is classification solely based on ImageNet?

To answer this question, we consider the following approaches where a model learned on ImageNet is applied to our dataset.

- **CNN-M-128**: We deploy the CNN-M-128 architecture of [16] that was used for feature extraction. However, instead of using the activations of the penultimate layer, we use the probability outputs of the final layer. If the probability of a certain animal is greater than 0.5, we conclude that the animal is present in the frame. The precision and the recall of this method are rather low. The reason is largely attributed to the domain shift (Figure 5)- the background plays a bigger role in our video, compared to the ImageNet images where the subject is central. Moreover, the images in ImageNet are of better quality with a high resolution, while the video key frames are of lower quality. Additionally, only 15 of our 19 names were present in the 1000 classes of ImageNet. However, the drop in recall resulting from the 4 missing classes was only 5.15%.

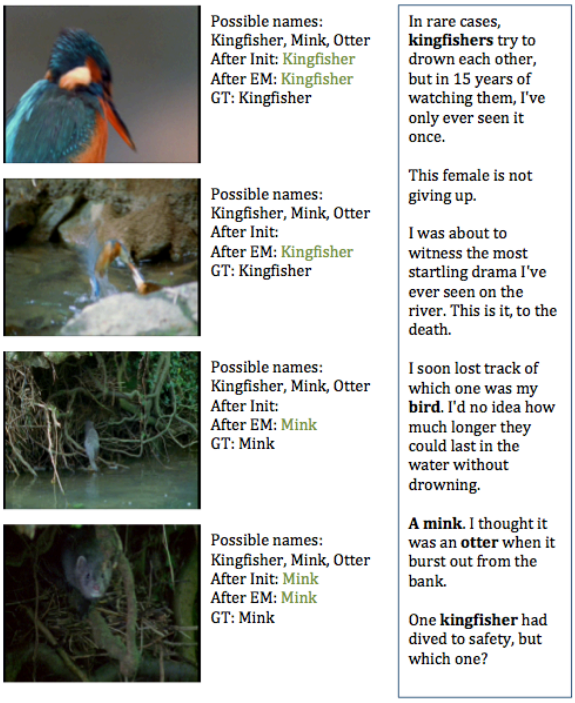

Figure 4: Annotating animals shown in the video key frames using the subtitle: Key frames (left), Guessed names (center), Subtitle (right); Init refers to the initialization using ImageNet and GT refers to the ground truth

- **CNN-M-128 filtered**: We modified the above approach so as to exclude those animals that were not in our dataset. Although this has lead to an increase in the precision compared to the above approach, the

Table 1: Evaluation of the indexing of frames (in %)

Method	Prec.	Rec.	F_1
CNN-M-128	13.1	25.8	17.3
CNN-M-128 filtered	55.0	25.8	35.1
ImageNet SVM_{raw}	32.9	28.9	30.8
Only text	42.5	**97.7**	59.3
Clustering + text + EM	55.7	36.4	44.0
ImageNet $SVM_{binarized}$	28.9	29.9	29.4
Binarization + text + EM	55.9	44.6	49.6
ImageNet NBC	15.7	45.5	23.4
ImageNet NBC + text	**57.6**	44.6	50.3
ImageNet NBC + text + EM	**57.3**	**88.7**	**69.6**

Figure 5: Images of crocodile from ImageNet (left) and keyframes containing crocodile (right)

recall remains low.

- **ImageNet SVM_{raw}**: Yet another simple solution to the problem of labeling is to train SVMs on labeled data, and apply it on our unlabeled dataset. Here, we train SVMs on the images extracted from ImageNet (in a one-vs-rest scheme), using the raw (non-binarized) CNN features, and test it on our dataset on the raw visual features. Note that the precision, recall and consequently the F_1 measure are quite low. Again, this is a result of the domain shift.

6.2. *How good is the text?*

We consider the baseline **Only text** which basically assigns all the possible names derived from 5 subtitles to the left and right of the frame. Note that the precision is quite low. However, the recall is very high; by extracting the names within this range of subtitles, almost all animal mentions were recovered. A recall of 100% is not achieved owing to two reasons - First, there were a couple of frames showing ducks, but the word duck is not mentioned in the subtitles over the entire video. Second, because we use a subtitle window of 5 subtitles, a few names are missed.

6.3. *Will clustering-based solutions work?*

The results of the labeling in an ideal scenario with manually annotated bounding boxes, as described in Section 3, using the approach of [4] with CNN features are shown in Table 2. *Annotation* indicates which bounding box in the video maps to which entity in the subtitle. *Frame indexing* indicates what animals shown in the frame are mapped to entities in the subtitles, considering the objects in the video frame and the entities in the subtitles as two distinct groups. The frame indexing deals with the mapping of the groups of objects in the frame to the groups of entities in the subtitles, ignoring the actual correspondence of the individual animals/entities.

We also apply the approach of [4] to entire frames rather than bounding boxes[4]. This is the baseline **Clustering + text + EM**. Here, we used k-means clustering to cluster the frames, with k set to 20, since there were 19 entities and we added 1 cluster for the background. Figure 6 shows some of the clusters obtained. It can be seen that the clusters are rather noisy. First, when there are multiple species in the same frame, they are forced into one cluster. For example, in the first cluster of Figure 6, zebra and crocodile are in the same cluster, simply because they are in the same frame. Second, even when frames with just one animal are involved, they are often grouped incorrectly. For example, in the first cluster of Figure 6, hippopotamus, crocodile, and kingfisher are in the same cluster. As a result, the performance of this approach in our setting is low, especially, the recall. This is because when a frame falls into the wrong cluster, the likelihood of the associated name would be very low, based on the other elements of the cluster. Next, we show that the binarization we proposed copes with this issue.

[4]There exist methods such as [27] to propose bounding boxes. While these methods have a high recall, the precision is often not sufficient for methods such as [4] to work. We experimented with the top 1 and 2 bounding boxes per frame, and found the performance quite low.

Table 2: Clustering-based algorithm applied on manually annotated bounding boxes (in %)

Method	Annotation	Frame Indexing		
	$F1$	Prec.	Rec.	$F1$
Initialization	80.80	87.1	82.3	84.6
After EM	83.80	88.5	86.4	87.4
Ground truth clusters	95.1	96.6	95.2	95.9

6.4. What is the impact of binarization?

To study the impact of binarization, we consider the following baselines.

- **ImageNet SVM**$_{binarized}$: We train SVMs as in Section 6.1, but using the binarized features instead of the raw features. The classifiers are then applied to our dataset with binarized features. While the results of this approach are quite low, they are comparable to the case with raw features (ImageNet SVM$_{raw}$). This is consistent with the study of [10].

- **Binarization + text + EM**: In this approach, rather than clustering the entire frames, we binarize[5] features along the dimensions. We start with uniform probabilities of the binarized features for the names (instead of learning from ImageNet). These probabilities are then refined by the E and M steps denoted by Eq. 3 and 8 respectively. Note that the precision improves significantly over the clustering-based approach (Clustering + text + EM). There is also an improvement in recall. *Binarization as an approach to grouping seems better than clustering in this setup with CNN features.*

6.5. What is the value of the iterative learning?

To evaluate our pipeline, we consider the following approaches.

- **ImageNet NBC**: In this approach, we learn initial probabilities (of the binarized features) from ImageNet and combine them using a Naive Bayes construction (Eq. 3). Textual cues are not used. It is interesting to compare the Naive Bayes (binarized) with the binarized SVM. While the precision of the Naive Bayes is quite low, the recall is better than that of SVM$_{binarized}$.

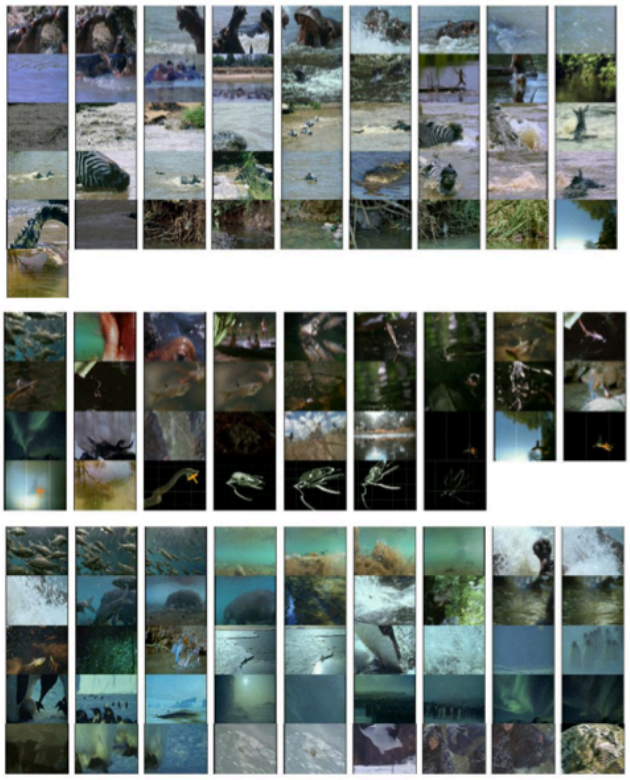

Figure 6: Clusters of key frames

- **ImageNet NBC + text**: As before, we learn initial probabilities (of the binarized features) from ImageNet and combine them using a Naive Bayes construction (Eq. 3). Further, we filter the labels by using the subtitles connected to a frame. Basically, we assign the labels to the frame only if the Naive Bayes predicts the label and if the label is also present in the adjoining subtitle. Note that the precision increases significantly over the above approach when textual cues are used. This is explained as follows. *The text provides good cues about the presence of certain animals.* For instance, very often hippopotamus or crocodiles were classified as salmon, simply because of the presence of the water body in the background. Using the textual cues, it is possible to arrive

[5]While it is possible to split the data into more bins, we have empirically found that the optimal number of bins for these features is 2.

Figure 7: Examples of key frames annotated by our system compared to the ground truth annotations (GT)

at the conclusion that it is unlikely that a salmon is shown here. This increase in precision is accompanied by a small drop in recall. The reason is that in some cases, although the classifier predicted a certain label correctly, the text suggested that the label may not be relevant in the context.

- **ImageNet NBC + text + EM**: This is basically the entire pipeline. We start with classifiers trained on ImageNet and iteratively adapt them to our dataset by making use of the textual co-occurrence information. Compare these results to two other approaches

    1. **Binarization + text + EM**, where we started with uniform probabilities rather than learning from ImageNet. Note the *significant increase in the recall and precision when the probabilities are learned from ImageNet.* The probabilities learned from ImageNet provide a good initialization that is essential to make the method converge to a meaningful result.
    2. **ImageNet NBC + text**, where we learned probabilities from ImageNet and filtered the labels using the text. Again, there is an increase in the recall because the *EM iterations adapt the classifiers trained to suit our data.*

The learning from ImageNet combined with the iterative use of textual cues that suggest the relevance of certain animals has boosted the recall significantly.

In addition to the evaluation on the entire dataset, we divided the frames into chapters and executed the pipeline on the individual chapters. The macro-average precision, recall and F_1 were 55.6%, 92.2% and 69.4%, while the micro-average precision, recall and F_1 were 58.1%, 94.3% and 71.9% respectively. These results are in line with the finding that the entire pipeline improves over each of the other methods for our documentary dataset.

Additionally, we tested the statistical significance of the results using a frame-level paired t-test and found that our method was significantly better ($p < 0.001$) than all approaches. Note that we are interested in a method that has the best performance in terms of precision and recall taken together. Figure 7 shows some examples of key frames annotated by our system. Particularly, even though the image with the penguins (4^{th} key frame) is hazy, this algorithm is successful in identifying the correct animal. Our algorithm is also successful in deducing that there are no animals in a frame (Figure 7, 3^{rd} key frame).

7. Summary and Conclusions

This paper shows that *by training classifiers on an external labeled dataset, and adapting them iteratively to the target dataset, using textual cues, the accuracy of classification can be improved.* This is applied to the context of recognizing objects such as animals shown in the video with subtitles, in the absence of visual demarcators such as bounding boxes. Exploiting the synergy between the visual features, textual cues and an external dataset, the accuracy of our approach is significantly better than a) a purely vision-based approach or b) purely text-based approach or c) an approach that uses both text and vision, but without labeled examples or d) an approach that uses both text and vision, and labeled (out-of-domain) examples, but without the adaptive learning.

In future, we wish to apply our algorithm to other datasets for furtherance of the evaluation scope. Additionally, we would like to determine the influence of the back-

ground in the recognition of animal, to determine whether or not the background should be used. Further, we intend to filter out regions of no interest which would confuse the clustering or classification. Applying these techniques allows making videos 'searchable' by automatically indexing them.

Acknowledgements

We thank the three anonymous reviewers for their insightful comments. This work was funded by DBOF Grant DOF/12/043.

References

[1] T. L. Berg, A. C. Berg, J. Edwards, M. Maire, R. White, Y. W. Teh, E. G. Learned-Miller, D. A. Forsyth, Names and faces in the news, In *CVPR* 2004.

[2] P. T. Pham, T. Tuytelaars, M.-F. Moens, Naming people in news videos with label propagation, IEEE Multimedia 18(3), (2011), 44-55.

[3] M. Everingham, J. Sivic, A. Zisserman, "Hello! My name is... Buffy"–Automatic Naming of Characters in TV Video. In *BMVC* 2006.

[4] T. Dusart, A. Nurani Venkitasubramanian, M.-F. Moens, Cross-modal alignment for wildlife recognition, In 2^{nd} ACM international workshop on Multimedia analysis for ecological data, ACM, 2013.

[5] A. Karpathy, L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions, In *CVPR* 2015.

[6] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. Platt, et al., From captions to visual concepts and back, In *CVPR* 2015.

[7] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, K. Saenko, Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition, In *ICCV*, 2013.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, In *CVPR*, 2009.

[9] P. Koehn, J. Schroeder, Experiments in domain adaptation for statistical machine translation, In 2^{nd} Workshop on Statistical Machine Translation, ACL, 2007.

[10] Y. Li, L. Liu, C. Shen, A. v. d. Hengel, Mid-level deep pattern mining, In *CVPR* 2015.

[11] T. L. Berg, D. A. Forsyth, Animals on the web, In *CVPR*, 2006.

[12] H. M. Afkham, A. T. Targhi, J.-O. Eklundh, A. Pronobis, Joint visual vocabulary for animal classification, In *ICPR*, 2008.

[13] C. Schmid, Constructing models for content-based image retrieval, In *CVPR*, 2001.

[14] D. Ramanan, D. A. Forsyth, K. Barnard, Building models of animals from video, IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(8), (2006), 1319–1334.

[15] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes (VOC) Challenge, IJCV, 88(2), (2010), 303–338.

[16] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The caltech-ucsd birds-200-2011 dataset, Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[17] A. Khosla, N. Jayadevaprakash, B. Yao, L. Fei-Fei.: Novel dataset for fine-grained image categorization, In 1^{st} Workshop on Fine-Grained Visual Categorization, In *CVPR*, 2011.

[18] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, Return of the devil in the details: Delving deep into convolutional nets, In *BMVC* 2014.

[19] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Computation, 1(4), (1989), 541–551.

[20] Guillaumin, M., Mensink, T., Verbeek, J., Schmid, C. (2008, June). Automatic face naming with caption-based supervision. In *CVPR*, 2008.

[21] H. Daumé III, Frustratingly easy domain adaptation, In *ACL* 2007.

[22] K. Nigam, A. K. McCallum, S. Thrun, T. Mitchell, Text classification from labeled and unlabeled documents using EM, Machine Learning, 39(2-3), (2000), 103-134.

[23] A. Bergamo, L. Torresani, Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach, In *NIPS*, 2010.

[24] R. Gopalan, R. Li, R. Chellappa, Domain adaptation for object recognition: An unsupervised approach, In *ICCV*, 2011.

[25] Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T. (2013, December). Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, 2013.

[26] P. Hellier, V. Demoulin, L. Oisel, P. Perez, A contrario shot detection, In *ICIP*, 2012.

[27] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.